

# Flex static code analysis

Unique rules to find Bugs, Security Hotspots, and Code Smells in your FLEX code

1.	Security.allowDomain(...) should only be used in a tightly focused manner <a href="#">Vulnerability</a>
2.	The flash.system.Security.exactSettings property should never be set to false <a href="#">Vulnerability</a>
3.	Dynamic classes should not be used <a href="#">Code Smell</a>
4.	"LocalConnection" should be configured to narrowly specify the domains with which local connections to other Flex application are allowed <a href="#">Vulnerability</a>
5.	"default" clauses should be first or last <a href="#">Code Smell</a>
6.	Event types should be defined in metadata tags <a href="#">Code Smell</a>
7.	Event names should not be hardcoded in event listeners <a href="#">Code Smell</a>
8.	The special "star" type should not be used <a href="#">Code Smell</a>
9.	Variables of the "Object" type should not be used <a href="#">Code Smell</a>
10.	Methods should not be empty <a href="#">Code Smell</a>
11.	Constant names should comply with a naming convention <a href="#">Code Smell</a>
12.	All branches in a conditional structure should not have exactly the same implementation <a href="#">Bug</a>
13.	Classes that extend "Event" should override "Event.clone()" <a href="#">Bug</a>
14.	Constructors should not dispatch events <a href="#">Bug</a>
15.	"ManagedEvents" tags should have companion "Event" tags <a href="#">Bug</a>

16.	Objects should not be instantiated inside a loop <a href="#">Code Smell</a>
17.	Two branches in a conditional structure should not have exactly the same implementation <a href="#">Code Smell</a>
18.	Constructor bodies should be as lightweight as possible <a href="#">Code Smell</a>
19.	Only "while", "do" and "for" statements should be labelled <a href="#">Code Smell</a>
20.	Statements, operators and keywords specific to ActionScript 2 should not be used <a href="#">Code Smell</a>
21.	"for" loop stop conditions should be invariant <a href="#">Code Smell</a>
22.	Unused function parameters should be removed <a href="#">Code Smell</a>
23.	Unused "private" functions should be removed <a href="#">Code Smell</a>
24.	Local variables should not shadow class fields <a href="#">Code Smell</a>
25.	Nested blocks of code should not be left empty <a href="#">Code Smell</a>
26.	Functions should not have too many parameters <a href="#">Code Smell</a>
27.	Unused "private" fields should be removed <a href="#">Code Smell</a>
28.	Collapsible "if" statements should be merged <a href="#">Code Smell</a>
29.	Delivering code in production with debug features activated is security-sensitive <a href="#">Security Hotspot</a>
30.	Unused local variables should be removed <a href="#">Code Smell</a>
31.	Package definition should be separate from Class definition <a href="#">Code Smell</a>
32.	Constructors should not have a "void" return type

	<a href="#">Code Smell</a>
33.	"public static" fields should be constant <a href="#">Code Smell</a>
34.	"switch" statements should have at least 3 "case" clauses <a href="#">Code Smell</a>
35.	Package names should comply with a naming convention <a href="#">Code Smell</a>
36.	Overriding methods should do more than simply call the same method in the super class <a href="#">Code Smell</a>
37.	Public constants and fields initialized at declaration should be "const static" rather than merely "const" <a href="#">Code Smell</a>
38.	Local variable and function parameter names should comply with a naming convention <a href="#">Code Smell</a>
39.	Field names should comply with a naming convention <a href="#">Code Smell</a>
40.	Boolean literals should not be redundant <a href="#">Code Smell</a>
41.	Empty statements should be removed <a href="#">Code Smell</a>
42.	Class names should comply with a naming convention <a href="#">Code Smell</a>
43.	Function names should comply with a naming convention <a href="#">Code Smell</a>
44.	The element type of an array field should be specified <a href="#">Code Smell</a>
45.	Track lack of copyright and license headers <a href="#">Code Smell</a>
46.	Octal values should not be used <a href="#">Code Smell</a>
47.	Switch cases should end with an unconditional "break" statement <a href="#">Code Smell</a>
48.	"switch" statements should not be nested <a href="#">Code Smell</a>
49.	

	Cyclomatic Complexity of functions should not be too high <a href="#">Code Smell</a>
50.	Control flow statements "if", "for", "while" and "switch" should not be nested too deeply <a href="#">Code Smell</a>
51.	Cyclomatic Complexity of classes should not be too high <a href="#">Code Smell</a>
52.	"switch" statements should have "default" clauses <a href="#">Code Smell</a>
53.	Multiple "++" or "--" unary operators should not be used in a single arithmetic expression <a href="#">Bug</a>
54.	"with" statements should not be used <a href="#">Bug</a>
55.	Useless "if(true) {...}" and "if(false){...}" blocks should be removed <a href="#">Bug</a>
56.	Flex parser failure <a href="#">Code Smell</a>
57.	Classes should not have too many fields <a href="#">Code Smell</a>
58.	Classes should not have too many methods <a href="#">Code Smell</a>
59.	"===" and "!===" should be used instead of "==" and "!=" <a href="#">Code Smell</a>
60.	Track breaches of an XPath rule <a href="#">Code Smell</a>
61.	Functions should not have too many lines of code <a href="#">Code Smell</a>
62.	Sections of code should not be commented out <a href="#">Code Smell</a>
63.	Track comments matching a regular expression <a href="#">Code Smell</a>
64.	Statements should be on separate lines <a href="#">Code Smell</a>
65.	Public classes, methods, properties and metadata should be documented with ASDoc <a href="#">Code Smell</a>
66.	

	"switch case" clauses should not have too many lines of code <u>Code Smell</u>
67.	
	Functions should not contain too many return statements <u>Code Smell</u>
68.	
	Lines should not be too long <u>Code Smell</u>
69.	
	The "trace" function should not be used <u>Vulnerability</u>
70.	
	"Alert.show(...)" should not be used <u>Vulnerability</u>
71.	
	Cases in a "switch" should not have the same condition <u>Bug</u>
72.	
	Method visibility should be explicitly declared <u>Bug</u>
73.	
	"MovieClip.onEnterFrame" event handler should not be used <u>Code Smell</u>
74.	
	Statements should end with semicolons <u>Code Smell</u>
75.	
	Loggers should be "private static const" and should share naming convention <u>Code Smell</u>
76.	
	A function should have a single point of exit at the end of the function <u>Code Smell</u>